AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A method comprising:

identifying in a first sequence of instructions associated with a source architecture, said first sequence comprising one or more floating point control instructions, a rounding mode of one of said one or more floating point control instructions, and

translating the first sequence of instructions into a second sequence of instructions associated with a target architecture,

wherein said second sequence does not include a floating point control instruction and the translation of said second sequence is conditional on said rounding mode, and

wherein results produced by executing said second sequence on a processor that complies with said target architecture are substantially the same as results produced by executing said first sequence on a processor that complies with said source architecture.

2. (Original) The method of claim 1, wherein said second sequence includes an instruction to round an initial floating point number to a first floating point number using round to zero rounding mode, regardless of a rounding mode setting of the target architecture.

3. (Original) The method of claim 1, wherein said first sequence of instructions and said second sequence of instructions are binary code.

4. (Cancelled)

5. (Previously Presented) A method comprising:

translating a first sequence of instructions associated with a source architecture into a second sequence of instructions associated with a target architecture,

wherein said first sequence includes an instruction to save a first rounding mode, followed by an instruction to set a second rounding mode that is not round to zero, followed by an instruction to generate an integer number by rounding an initial floating point number according to said second rounding mode, followed by an instruction to set said first rounding mode,

wherein the translation of said second sequence is conditional on said second rounding mode and includes an instruction to generate a first floating point number by

rounding to zero said initial floating point number and an instruction to compare a value of said first floating point number to a value of said initial floating point number, and

wherein results produced by executing said second sequence on a processor that complies with said target architecture are substantially the same as results produced by executing said first sequence on a processor that complies with said source architecture.

6. (Cancelled)

7. (Currently Amended) The method of claim ~~7~~ 5, wherein said second rounding mode is a round toward positive infinity mode, and said second sequence further includes an instruction to add one to said first floating point number in the event that said first floating point number is not negative and the value of said first floating point number is not equal to the value of said initial floating point number.

8. (Currently Amended) The method of claim ~~7~~ 5, wherein said second rounding mode is a round toward negative infinity mode, and said second sequence further includes an instruction to subtract one from said first floating point number in the event that said first floating point number is not positive and the value of said first floating point number is not equal to the value of said second floating point number.

9. (Currently Amended) A method for converting a floating point representation to an integer representation with a round to negative infinity rounding mode in a computer, the method comprising:

using a round to zero forced mode instruction to convert an initial floating point number stored in a computer readable medium to a first floating point number stored in a computer readable medium; ~~and~~

if said first floating point number is not positive, and a value of said first floating point number is not equal to a value of said initial floating point number, generating a second floating point number by subtracting one from said first floating point number, and generating said integer representation stored in a computer readable medium by converting the floating point representation of said second floating point number to an integer representation[[,]] ; and

executing a computer instruction on a computer using said integer representation as an argument.

10. (Original) The method of claim 9, further comprising:

if said value of said first floating point number is equal to said value of said initial floating point number, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

11. (Original) The method of claim 9, further comprising:

if said first floating point number is positive, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

12. (Currently Amended) A method for converting a floating point representation to an integer representation with a round to positive infinity rounding mode in a computer, the method comprising:

using a round to zero forced mode instruction to convert an initial floating point number stored in a computer readable medium to a first integer number stored in a computer readable medium; ~~and~~

if said first floating point number is not negative, and a value of said first floating number is not equal to a value of said initial floating point number, generating a second floating point number stored in a computer readable medium by adding one to said first floating point number, and generating said integer representation by converting the floating point representation of said second floating point number to integer representation; and

executing a computer instruction on a computer using said integer representation as an argument.

13. (Original) The method of claim 12, further comprising:

if said value of said first floating point number is equal to said value of said initial floating point number, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

14. (Original) The method of claim 12, further comprising:

if said first floating point number is negative, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

15. (Previously Presented) An article comprising a storage medium having stored thereon instructions that, when executed by a computing platform, result in

identifying in a first sequence of instructions associated with a source architecture, said first sequence including one or more floating point control instructions, a rounding mode of one of said one or more floating point control instructions, and

translating the first sequence of instructions into a second sequence of instructions associated with a target architecture,

wherein said second sequence does not include a floating point control instruction and the translation of said second sequence is conditional on said rounding mode, and

wherein results produced by executing said second sequence on a processor that complies with said target architecture are substantially the same as results produced by executing said first sequence on a processor that complies with said source architecture.

16. (Original) The article of claim 15, wherein said second sequence includes an instruction to round an initial floating point number to a first floating point number using round to zero rounding mode, regardless of the rounding mode setting of the target architecture.

17. (Cancelled)

18. (Currently Amended) An article comprising a storage medium having stored thereon instructions for conversion of a floating point representation to an integer representation with a round to negative infinity rounding mode, wherein the instructions, when executed by a computing platform, result in:

using a round to zero forced mode instruction to convert an initial floating point number to a first floating point number; ~~and~~

if said first floating point number is not positive, and a value of said first floating point number is not equal to a value of said initial floating point number, generating a second floating point number by subtracting one from said first floating point number, and generating said integer representation by converting the floating point representation of said second floating point number to an integer representation; and

storing said integer in a computer readable medium as an argument to an instruction executed by a computer.

19. (Original) The article of claim 18, wherein the instructions further result in:

if said value of said first floating point number is equal to said value of said initial floating point number, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

20. (Original) The article of claim 19, wherein the instructions further result in:

if said first floating point number is positive, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

21. (Currently Amended) An article comprising a storage medium having stored thereon instructions for conversion of a floating point representation to an integer representation with a round to positive infinity rounding mode, wherein the instructions, when executed by a computing platform, result in:

using a round to zero forced mode instruction to convert an initial floating point number to a first floating point number; ~~and~~

if said first floating point number is not negative, and a value of said first floating point number is not equal to a value of said initial floating point number, generating a second floating point number by adding one to said first floating point number, and generating said integer representation by converting the floating point representation of said second floating point number to integer representation; and

storing said integer in a computer readable medium as an argument to an instruction executed by a computer.

22. (Original) The article of claim 21, wherein the instructions further result in:

if said value of said first floating point number is equal to said value of said initial floating point number, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

23. (Original) The article of claim 21, further comprising:

if said first floating point number is negative, generating said integer representation by converting the floating point representation of said first floating point number to an integer representation.

24. (Previously Presented) An apparatus comprising:

a memory device; and

a processor to identify in a first sequence of instructions associated with a source architecture, said first sequence including one or more floating point control instructions, a rounding mode of one of said one or more floating point control instructions, and to translate the first sequence of instructions into a second sequence of instructions associated with a target architecture,

wherein said second sequence does not include a floating point control instruction and the translation of said second sequence is conditional on said rounding mode, and

wherein results produced by executing said second sequence on a processor that complies with said target architecture are substantially the same as results produced by executing said first sequence on a processor that complies with said source architecture.

25. (Original) The apparatus of claim 24, wherein said processor is to translate said first sequence into said second sequence so that said second sequence includes an instruction to round an initial floating point number to a first floating point number using round to zero rounding mode, regardless of the rounding mode setting of the target architecture.

26. (Cancelled)